Intel486™ Processor Family
Programmer's Reference Manual

Intel486™ MICROPROCESSOR FAMILY PROGRAMMER'S REFERENCE MANUAL

1995

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION TO THE
Intel486™ PROCESSOR FAMILY **Page**

1.1 ORGANIZATION OF THIS MANUAL 1-2
1.1.1 Part I—Application Programming 1-3
1.1.2 Part II—System Programming 1-3
1.1.3 Part III—Numeric Processing 1-4
1.1.4 Part IV—Compatibility 1-5
1.1.5 Part V—Instruction Set 1-6
1.1.6 Appendices 1-6
1.2 RELATED LITERATURE 1-6
1.3 NOTATIONAL CONVENTIONS 1-6
1.3.1 Bit and Byte Order 1-7
1.3.2 Undefined Bits and Software Compatibility 1-7
1.3.3 Instruction Operands 1-8
1.3.4 Hexadecimal Numbers 1-8
1.3.5 Segmented Addressing 1-9
1.3.6 Exceptions 1-9

CHAPTER 2
BASIC PROGRAMMING MODEL

2.1 MEMORY ORGANIZATION 2-1
2.1.1 Unsegmented or "Flat" Model 2-3
2.1.2 Segmented Model 2-3
2.2 DATA TYPES 2-3
2.3 REGISTERS 2-6
2.3.1 General Registers 2-8
2.3.2 Segment Registers 2-9
2.3.3 Stack Implementation 2-11
2.3.4 Flags Register 2-12
2.3.4.1 STATUS FLAGS 2-13
2.3.4.2 CONTROL FLAG 2-14
2.3.4.3 INSTRUCTION POINTER 2-14
2.4 INSTRUCTION FORMAT 2-15
2.5 OPERAND SELECTION 2-16
2.5.1 Immediate Operands 2-17
2.5.2 Register Operands 2-18
2.5.3 Memory Operands 2-18
2.5.3.1 SEGMENT SELECTION 2-19
2.5.3.2 EFFECTIVE-ADDRESS COMPUTATION 2-20
2.6 INTERRUPTS AND EXCEPTIONS 2-22

CHAPTER 3
APPLICATION PROGRAMMING

3.1 DATA MOVEMENT INSTRUCTIONS 3-1
3.1.1 General-Purpose Data Movement Instructions 3-1
3.1.2 Stack Manipulation Instructions 3-2
3.1.3 Type Conversion Instructions 3-3

	Page
3.2 BINARY ARITHMETIC INSTRUCTIONS	3-6
3.2.1 Addition and Subtraction Instructions	3-7
3.2.2 Comparison and Sign Change Instruction	3-7
3.2.3 Multiplication Instructions	3-8
3.2.4 Division Instructions	3-9
3.3 DECIMAL ARITHMETIC INSTRUCTIONS	3-9
3.3.1 Packed BCD Adjustment Instructions	3-10
3.3.2 Unpacked BCD Adjustment Instructions	3-10
3.4 LOGICAL INSTRUCTIONS	3-11
3.4.1 Boolean Operation Instructions	3-11
3.4.2 Bit Test and Modify Instructions	3-11
3.4.3 Bit Scan Instructions	3-12
3.4.4 Shift and Rotate Instructions	3-12
3.4.4.1 SHIFT INSTRUCTIONS	3-12
3.4.4.2 DOUBLE-SHIFT INSTRUCTIONS	3-15
3.4.4.3 ROTATE INSTRUCTIONS	3-16
3.4.4.4 FAST "bit blt" USING DOUBLE-SHIFT INSTRUCTIONS	3-17
3.4.4.5 FAST BIT STRING INSERT AND EXTRACT	3-18
3.4.5 Byte-Set-On-Condition Instructions	3-21
3.4.6 Test Instruction	3-22
3.5 CONTROL TRANSFER INSTRUCTIONS	3-22
3.5.1 Unconditional Transfer Instructions	3-22
3.5.1.1 JUMP INSTRUCTION	3-22
3.5.1.2 CALL INSTRUCTIONS	3-23
3.5.1.3 RETURN AND RETURN-FROM-INTERRUPT INSTRUCTIONS	3-23
3.5.2 Conditional Transfer Instructions	3-24
3.5.2.1 CONDITIONAL JUMP INSTRUCTIONS	3-24
3.5.2.2 LOOP INSTRUCTIONS	3-24
3.5.2.3 EXECUTING A LOOP OR REPEAT ZERO TIMES	3-25
3.5.3 Software Interrupts	3-25
3.6 STRING OPERATIONS	3-26
3.6.1 Repeat Prefixes	3-27
3.6.2 Indexing and Direction Flag Control	3-28
3.6.3 String Instructions	3-28
3.7 INSTRUCTIONS FOR BLOCK-STRUCTURED LANGUAGES	3-29
3.8 FLAG CONTROL INSTRUCTIONS	3-34
3.8.1 Carry and Direction Flag Control Instructions	3-34
3.8.2 Flag Transfer Instructions	3-36
3.9 NUMERIC INSTRUCTIONS	3-37
3.10 SEGMENT REGISTER INSTRUCTIONS	3-37
3.10.1 Segment-Register Transfer Instructions	3-38
3.10.2 Far Control Transfer Instructions	3-38
3.10.3 Data Pointer Instructions	3-39
3.11 MISCELLANEOUS INSTRUCTIONS	3-40
3.11.1 CPU_ID Detection Code	3-40
3.11.2 Address Calculation Instruction	3-40
3.11.3 No-Operation Instruction	3-41
3.11.4 Translate Instruction	3-41
3.11.5 Byte Swap Instruction	3-41
3.11.6 Exchange-and-Add Instruction	3-41
3.11.7 Compare-and-Exchange Instruction	3-41

CHAPTER 4
SYSTEM ARCHITEC

4.1 SYSTEM REGIS
4.1.1 System Flags
4.1.2 Memory-Manag
4.1.3 Control Registe
4.1.4 Debug Registe
4.1.5 Test Registers

CHAPTER 5
MEMORY MANAGE

5.1 SELECTING A S
5.1.1 Flat Model
5.1.2 Protected Flat
5.1.3 Multi-Segment
5.2 SEGMENT TRA
5.2.1 Segment Regi
5.2.2 Segment Sele
5.2.3 Segment Desc
5.2.4 Segment Desc
5.2.5 Descriptor Tab
5.3 Page Translatio
5.3.1 PG Bit Enable
5.3.2 Linear Addres
5.3.3 Page Tables
5.3.4 Page-Table E
5.3.4.1 PAGE FRA
5.3.4.2 PRESENT
5.3.4.3 ACCESSED
5.3.4.4 READ/WRI
5.3.4.5 PAGE-LEV
5.3.5 Translation L
5.4 COMBINING S
5.4.1 Flat Model
5.4.2 Segments Sp
5.4.3 Pages Spann
5.4.4 Non-Aligned
5.4.5 Aligned Page
5.4.6 Page-Table

CHAPTER 6
PROTECTION

6.1 SEGMENT-LE
6.2 SEGMENT DE
6.2.1 Type Checki
6.2.2 Limit Checki
6.2.3 Privilege Le
6.3 RESTRICTIN
6.3.1 Accessing D

Page

CHAPTER 4
SYSTEM ARCHITECTURE

4.1 SYSTEM REGISTERS 4-1
4.1.1 System Flags 4-2
4.1.2 Memory-Management Registers 4-4
4.1.3 Control Registers 4-5
4.1.4 Debug Registers 4-9
4.1.5 Test Registers 4-9

CHAPTER 5
MEMORY MANAGEMENT

5.1 SELECTING A SEGMENTATION MODEL 5-2
5.1.1 Flat Model 5-3
5.1.2 Protected Flat Model 5-4
5.1.3 Multi-Segment Model 5-5
5.2 SEGMENT TRANSLATION 5-6
5.2.1 Segment Registers 5-8
5.2.2 Segment Selectors 5-9
5.2.3 Segment Descriptors 5-10
5.2.4 Segment Descriptor Tables 5-14
5.2.5 Descriptor Table Base Registers 5-16
5.3 Page Translation 5-16
5.3.1 PG Bit Enables Paging 5-18
5.3.2 Linear Address 5-18
5.3.3 Page Tables 5-18
5.3.4 Page-Table Entries 5-19
5.3.4.1 PAGE FRAME ADDRESS 5-19
5.3.4.2 PRESENT BIT 5-20
5.3.4.3 ACCESSED AND DIRTY BITS 5-21
5.3.4.4 READ/WRITE AND USER/SUPERVISOR BITS 5-21
5.3.4.5 PAGE-LEVEL CACHE CONTROL BITS 5-21
5.3.5 Translation Lookaside Buffer 5-22
5.4 COMBINING SEGMENT AND PAGE TRANSLATION 5-22
5.4.1 Flat Model 5-22
5.4.2 Segments Spanning Several Pages 5-23
5.4.3 Pages Spanning Several Segments 5-23
5.4.4 Non-Aligned Page and Segment Boundaries 5-23
5.4.5 Aligned Page and Segment Boundaries 5-24
5.4.6 Page-Table Per Segment 5-24

CHAPTER 6
PROTECTION

6.1 SEGMENT-LEVEL PROTECTION 6-1
6.2 SEGMENT DESCRIPTORS AND PROTECTION 6-2
6.2.1 Type Checking 6-3
6.2.2 Limit Checking 6-4
6.2.3 Privilege Levels 6-5
6.3 RESTRICTING ACCESS TO DATA 6-7
6.3.1 Accessing Data in Code Segments 6-7

	Page
6.4 RESTRICTING CONTROL TRANSFERS	6-8
6.5 GATE DESCRIPTORS	6-10
6.5.1 Stack Switching	6-14
6.5.2 Returning from a Procedure	6-16
6.6 INSTRUCTIONS RESERVED FOR THE OPERATING SYSTEM	6-18
6.6.1 Privileged Instructions	6-18
6.6.2 Sensitive Instructions	6-18
6.7 INSTRUCTIONS FOR POINTER VALIDATION	6-19
6.7.1 Descriptor Validation	6-20
6.7.2 Pointer Integrity and RPL	6-21
6.8 PAGE-LEVEL PROTECTION	6-21
6.8.1 Page-Table Entries Hold Protection Parameters	6-22
6.8.1.1 RESTRICTING ADDRESSABLE DOMAIN	6-22
6.8.1.2 TYPE CHECKING	6-22
6.8.2 Combining Protection of Both Levels of Page Tables	6-23
6.8.3 Overrides to Page Protection	6-23
6.9 COMBINING PAGE AND SEGMENT PROTECTION	6-23

CHAPTER 7
MULTITASKING

7.1 TASK STATE SEGMENT	7-2
7.2 TSS DESCRIPTOR	7-2
7.3 TASK REGISTER	7-4
7.4 TASK GATE DESCRIPTOR	7-6
7.5 TASK SWITCHING	7-7
7.6 TASK LINKING	7-9
7.6.1 Busy Bit Prevents Loops	7-12
7.6.2 Modifying Task Linkages	7-12
7.7 TASK ADDRESS SPACE	7-12
7.7.1 Task Linear-to-Physical Space Mapping	7-13
7.7.2 Task Logical Address Space	7-13

CHAPTER 8
INPUT/OUTPUT

8.1 I/O ADDRESSING	8-1
8.1.1 I/O Address Space	8-2
8.1.2 Memory-Mapped I/O	8-3
8.2 I/O INSTRUCTIONS	8-4
8.2.1 Register I/O Instructions	8-4
8.2.2 Block I/O Instructions	8-5
8.3 PROTECTION AND I/O	8-5
8.3.1 I/O Privilege Level	8-6
8.3.2 I/O Permission Bit Map	8-6

CHAPTER 9
EXCEPTIONS AND INTERRUPTS

9.1 EXCEPTION AND INTERRUPT VECTORS	9-1
9.2 INSTRUCTION RESTART	9-2
9.3 ENABLING AND DISABLING INTERRUPTS	9-3

9.3.1 NMI Masks F
9.3.2 IF Masks INT
9.3.3 RF Masks De
9.3.4 MOV or POP
9.4 PRIORITY AM
9.5 INTERRUPT D
9.6 IDT DESCRIPT
9.7 INTERRUPT T
9.7.1 Interrupt Pro
9.7.1.1 STACK OP
9.7.1.2 RETURNIN
9.7.1.3 FLAG USA
9.7.1.4 PROTECT
9.7.2 Interrupt Tas
9.8 ERROR CODE
9.9 EXCEPTION C
9.9.1 Interrupt 0—
9.9.2 Interrupt 1—
9.9.3 Interrupt 3—
9.9.4 Interrupt 4—
9.9.5 Interrupt 5—
9.9.6 Interrupt 6—
9.9.7 Interrupt 7—
9.9.8 Interrupt 8—
9.9.9 Interrupt 9—
9.9.10 Interrupt 10
9.9.11 Interrupt 11
9.9.12 Interrupt 12
9.9.13 Interrupt 13
9.9.14 Interrupt 1
9.9.14.1 PAGE FA
9.9.14.2 PAGE FA
9.9.15 Interrupt 1
9.9.16 Interrupt 1
9.10 EXCEPTION
9.11 ERROR CO

CHAPTER 10
INITIALIZATION

10.1 PROCESSO
10.2 INITIALIZAT
OVERDRIVE PR
10.3 SOFTWARE
10.3.1 System Ta
10.3.2 NMI Interr
10.3.3 First Instr
10.3.4 Enabling
10.4 SWITCHIN
10.4.1 System T
10.4.2 NMI Inter
10.4.3 PE Bit ....
10.5 SOFTWAR

	Page
9.3.1 NMI Masks Further NMIs	9-3
9.3.2 IF Masks INTR	9-3
9.3.3 RF Masks Debug Faults	9-4
9.3.4 MOV or POP to SS Masks Some Exceptions and Interrupts	9-4
9.4 PRIORITY AMONG SIMULTANEOUS EXCEPTIONS AND INTERRUPTS	9-5
9.5 INTERRUPT DESCRIPTOR TABLE	9-5
9.6 IDT DESCRIPTORS	9-7
9.7 INTERRUPT TASKS AND INTERRUPT PROCEDURES	9-7
9.7.1 Interrupt Procedures	9-7
9.7.1.1 STACK OF INTERRUPT PROCEDURE	9-7
9.7.1.2 RETURNING FROM AN INTERRUPT PROCEDURE	9-7
9.7.1.3 FLAG USAGE BY INTERRUPT PROCEDURE	9-8
9.7.1.4 PROTECTION IN INTERRUPT PROCEDURES	9-10
9.7.2 Interrupt Tasks	9-11
9.8 ERROR CODE	9-11
9.9 EXCEPTION CONDITIONS	9-12
9.9.1 Interrupt 0—Divide Error	9-13
9.9.2 Interrupt 1—Debug Exceptions	9-13
9.9.3 Interrupt 3—Breakpoint	9-14
9.9.4 Interrupt 4—Overflow	9-14
9.9.5 Interrupt 5—Bounds Check	9-14
9.9.6 Interrupt 6—Invalid Opcode	9-14
9.9.7 Interrupt 7—Device Not Available	9-15
9.9.8 Interrupt 8—Double Fault	9-16
9.9.9 Interrupt 9—(Intel reserved. Do not use.)	9-17
9.9.10 Interrupt 10—Invalid TSS	9-17
9.9.11 Interrupt 11—Segment Not Present	9-18
9.9.12 Interrupt 12—Stack Exception	9-19
9.9.13 Interrupt 13—General Protection	9-19
9.9.14 Interrupt 14—Page Fault	9-21
9.9.14.1 PAGE FAULT DURING TASK SWITCH	9-21
9.9.14.2 PAGE FAULT WITH INCONSISTENT STACK POINTER	9-22
9.9.15 Interrupt 16—Floating-Point Error	9-23
9.9.16 Interrupt 17—Alignment Check	9-23
9.10 EXCEPTION SUMMARY	9-24
9.11 ERROR CODE SUMMARY	9-24

CHAPTER 10
INITIALIZATION

10.1 PROCESSOR STATE AFTER RESET	10-1
10.2 INITIALIZATION REQUIREMENTS FOR THE FLOATING-POINT UNIT OF THE INTEL OVERDRIVE PROCESSORS	10-3
10.3 SOFTWARE INITIALIZATION IN REAL-ADDRESS MODE	10-5
10.3.1 System Tables	10-5
10.3.2 NMI Interrupt	10-5
10.3.3 First Instruction	10-5
10.3.4 Enabling Caching	10-5
10.4 SWITCHING TO PROTECTED MODE	10-6
10.4.1 System Tables	10-6
10.4.2 NMI Interrupt	10-6
10.4.3 PE Bit	10-6
10.5 SOFTWARE INITIALIZATION IN PROTECTED MODE	10-7

	Page
10.5.1 Segmentation	10-7
10.5.2 Paging	10-7
10.5.3 Tasks	10-8
10.6 TLB TESTING	10-8
10.6.1 Structure of the TLB	10-8
10.6.2 Test Registers	10-9
10.6.3 Test Operations	10-11
10.7 CACHE TESTING	10-12
10.7.1 Structure of the Cache	10-12
10.7.1.1 8K-bytes Internal Cache Structure	10-12
10.7.1.2 16K-bytes Internal Cache Structure	10-13
10.7.2 Test Registers	10-14
10.7.2.1 Additional Cache Testing Features for the Write-Back Mode	10-15
10.7.3 Test Operations	10-16
10.8 INITIALIZATION EXAMPLE	10-18

CHAPTER 11
DEBUGGING

11.1 DEBUGGING SUPPORT	11-1
11.2 DEBUG REGISTERS	11-2
11.2.1 Debug Address Registers (DR0-DR3)	11-2
11.2.2 Debug Control Register (DR7)	11-2
11.2.3 Debug Status Register (DR6)	11-4
11.2.4 Breakpoint Field Recognition	11-4
11.3 DEBUG EXCEPTIONS	11-5
11.3.1 Interrupt 1—Debug Exceptions	11-6
11.3.1.1 INSTRUCTION-BREAKPOINT FAULT	11-6
11.3.1.2 DATA-BREAKPOINT TRAP	11-7
11.3.1.3 GENERAL-DETECT FAULT	11-7
11.3.1.4 SINGLE-STEP TRAP	11-7
11.3.1.5 TASK-SWITCH TRAP	11-8
11.3.2 Interrupt 3—Breakpoint Instruction	11-8

CHAPTER 12
CACHING

12.1 INTRODUCTION TO CACHING	12-1
12.2 OPERATION OF THE INTERNAL CACHE	12-2
12.2.1 Cache Disabling Bits	12-2
12.2.2 Cache Management Instructions	12-3
12.2.3 Self-Modifying Code	12-4
12.3 PAGE-LEVEL CACHE MANAGEMENT	12-4
12.3.1 Cache Management Bits	12-4
12.3.1.1 PCD BIT	12-4
12.3.1.2 PWT BIT	12-5

CHAPTER 13
MULTIPROCESSING

13.1 LOCKED AND PSEUDO-LOCKED BUS CYCLES	13-1
13.1.1 LOCK Prefix and the LOCK# Signal	13-2

13.1.2 Automatic Loc
13.1.3 Pseudo-Lockin

CHAPTER 14
POWER MANAGEMI

14.1 SL ENHANCED
14.1.1 Introduction to
14.1.2 Terminology
14.1.3 SL Enhanced
14.1.3.1 SMRAM ....
14.1.3.2 SMRAM ST
14.1.3.3 EXIT FROM
14.1.4 System Mana
14.1.4.1 SMM ENTR
14.1.4.2 PROCESSO
14.1.4.3 EXECUTIN
14.1.4.4 EXCEPTIO
14.1.5 SMM Feature
14.1.5.1 SMM REVI
14.1.5.2 HALT AUTO
14.1.5.3 SMM BASE
14.1.6 Intel486 Proc
14.1.6.1 SMM CODE
14.1.6.2 EXCEPTIO
14.1.6.3 HALT DUR
14.1.6.4 RELOCATI

CHAPTER 15
INTRODUCTION TO

15.1 HISTORY .....
15.2 PERFORMAN
15.3 EASE OF USE
15.4 APPLICATION
15.5 PROGRAMMI

CHAPTER 16
ARCHITECTURE O

16.1 NUMERICAL
16.1.1 The FPU Re
16.1.2 The FPU St
16.1.3 Control Wor
16.1.4 The FPU Ta
16.1.5 Opcode Fiel
16.1.6 The Numeri
16.2 COMPUTATIO
16.2.1 Number Sys
16.2.2 Data Types
16.2.2.1 BINARY I
16.2.2.2 DECIMAL

Page

13.1.2 Automatic Locking 13-2
13.1.3 Pseudo-Locking 13-3

CHAPTER 14
POWER MANAGEMENT

14.1 SL ENHANCED Intel486™ PROCESSORS POWER MANAGEMENT 14-1
14.1.1 Introduction to System Management Mode Architecture 14-1
14.1.2 Terminology 14-1
14.1.3 SL Enhanced Intel486 Processor System Management Interrupt Processing 14-2
14.1.3.1 SMRAM 14-2
14.1.3.2 SMRAM STATE SAVE MAP 14-3
14.1.3.3 EXIT FROM SMM 14-5
14.1.4 System Management Mode Programming Model 14-6
14.1.4.1 SMM ENTRY 14-6
14.1.4.2 PROCESSOR ENVIRONMENT 14-7
14.1.4.3 EXECUTING SYSTEM MANAGEMENT MODE HANDLER 14-8
14.1.4.4 EXCEPTIONS AND INTERRUPTS WITHIN SMM 14-8
14.1.5 SMM Features 14-9
14.1.5.1 SMM REVISION IDENTIFIER 14-9
14.1.5.2 HALT AUTO RESTART 14-9
14.1.5.3 SMM BASE RELOCATION 14-10
14.1.6 Intel486 Processor SMM - Software Considerations 14-12
14.1.6.1 SMM CODE CONSIDERATIONS 14-12
14.1.6.2 EXCEPTION HANDLING 14-12
14.1.6.3 HALT DURING SMM 14-13
14.1.6.4 RELOCATING SMRAM TO AN ADDRESS ABOVE ONE MEGABYTE 14-13

CHAPTER 15
INTRODUCTION TO NUMERIC APPLICATIONS

15.1 HISTORY 15-1
15.2 PERFORMANCE 15-2
15.3 EASE OF USE 15-3
15.4 APPLICATIONS 15-4
15.5 PROGRAMMING INTERFACE 15-5

CHAPTER 16
ARCHITECTURE OF THE FLOATING-POINT UNIT

16.1 NUMERICAL REGISTERS 16-1
16.1.1 The FPU Register Stack 16-1
16.1.2 The FPU Status Word 16-2
16.1.3 Control Word 16-5
16.1.4 The FPU Tag Word 16-6
16.1.5 Opcode Field of Last Instruction 16-7
16.1.6 The Numeric Instruction and Data Pointers 16-8
16.2 COMPUTATION FUNDAMENTALS 16-9
16.2.1 Number System 16-10
16.2.2 Data Types and Formats 16-12
16.2.2.1 BINARY INTEGERS 16-13
16.2.2.2 DECIMAL INTEGERS 16-13

Page
16.2.2.3 REAL NUMBERS 16-13
16.2.3 Rounding Control 16-16
16.2.4 Precision Control 16-16

CHAPTER 17
SPECIAL COMPUTATIONAL SITUATIONS

17.1 SPECIAL NUMERIC VALUES 17-1
17.1.1 Denormal Real Numbers 17-1
17.1.1.1 DENORMALS AND GRADUAL UNDERFLOW 17-4
17.1.2 Zeros 17-6
17.1.3 Infinity 17-8
17.1.4 NaN (Not-a-Number) 17-8
17.1.4.1 SIGNALING NaNs 17-10
17.1.4.2 QUIET NaNs 17-11
17.1.5 Indefinite 17-12
17.1.6 Encoding of Data Types 17-12
17.1.7 Unsupported Formats 17-13
17.2 NUMERIC EXCEPTIONS 17-17
17.2.1 Handling Numeric Exceptions 17-18
17.2.1.1 AUTOMATIC EXCEPTION HANDLING 17-18
17.2.1.2 SOFTWARE EXCEPTION HANDLING 17-19
17.2.2 Invalid Operation 17-20
17.2.2.1 STACK EXCEPTION 17-20
17.2.2.2 INVALID ARITHMETIC OPERATION 17-21
17.2.3 Division by Zero 17-21
17.2.4 Denormal Operand 17-22
17.2.5 Numeric Overflow and Underflow 17-23
17.2.5.1 OVERFLOW 17-23
17.2.5.2 UNDERFLOW 17-24
17.2.6 Inexact (Precision) 17-26
17.2.7 Exception Priority 17-26
17.2.8 Standard Underflow/Overflow Exception Handler 17-27

CHAPTER 18
FLOATING-POINT INSTRUCTION SET

18.1 SOURCE AND DESTINATION OPERANDS 18-1
18.2 DATA TRANSFER INSTRUCTIONS 18-2
18.3 NONTRANSCENDENTAL INSTRUCTIONS 18-2
18.4 COMPARISON INSTRUCTIONS 18-4
18.5 TRANSCENDENTAL INSTRUCTIONS 18-5
18.6 CONSTANT INSTRUCTIONS 18-6
18.7 CONTROL INSTRUCTIONS 18-7

CHAPTER 19
NUMERIC APPLICATIONS

19.1 PROGRAMMING FACILITIES 19-1
19.1.1 High-Level Languages 19-1
19.1.2 C Programs 19-1
19.1.3 PL/M-386/486 19-2

19.1.4 ASM386/486 ...
19.1.4.1 DEFINING DA
19.1.4.2 RECORDS AI
19.1.4.3 Addressing M
19.1.5 Comparative Pr
19.2 CONCURRENT P
19.2.1 Managing Conc
19.2.1.1 INCORRECT
19.2.1.2 PROPER EX
19.3 NUMERIC EXCE
19.3.1 Usage Of The '

CHAPTER 20
SYSTEM-LEVEL CON

20.1 ARCHITECTURE
20.1.1 Independent of
20.2 PROCESSOR IN
20.2.1 System Initializ
20.2.2 Configuring the
20.2.3 Initializing the
20.2.3.1 Intel486 DX
20.2.3.2 Intel486 SX
20.2.4 Handling Num
20.2.5 Simultaneous
20.2.6 Exception Rec

CHAPTER 21
NUMERIC PROGRA

21.1 CONDITIONAL
21.2 EXCEPTION H
21.3 FLOATING-PO
21.3.1 Function Part
21.3.2 Exception Co
21.3.3 Special Instru
21.3.4 Description o
21.3.5 Scaling the V
21.3.5.1 INACCURA
21.3.5.2 AVOIDING
21.3.5.3 FINAL ADJ
21.3.6 Output Form
21.4 TRIGONOMET

CHAPTER 22
EXECUTING 286 A
Intel386™ DX OR S

22.1 TWO WAYS T
22.2 DIFFERENCE
22.2.1 Wraparound
22.2.2 Reserved W

Page

19.1.4 ASM386/486 19-3
19.1.4.1 DEFINING DATA 19-4
19.1.4.2 RECORDS AND STRUCTURES 19-5
19.1.4.3 Addressing Methods 19-5
19.1.5 Comparative Programming Example 19-6
19.2 CONCURRENT PROCESSING 19-11
19.2.1 Managing Concurrency 19-12
19.2.1.1 INCORRECT EXCEPTION SYNCHRONIZATION 19-14
19.2.1.2 PROPER EXCEPTION SYNCHRONIZATION 19-14
19.3 NUMERIC EXCEPTIONS: INTERRUPT SAMPLING WINDOW 19-14
19.3.1 Usage Of The "No-Wait" Floating-Point Instruction Class 19-15

CHAPTER 20
SYSTEM-LEVEL CONSIDERATIONS

20.1 ARCHITECTURE 20-1
20.1.1 Independent of Addressing Mode 20-1
20.2 PROCESSOR INITIALIZATION AND CONTROL 20-1
20.2.1 System Initialization 20-2
20.2.2 Configuring the Numerics Environment 20-2
20.2.3 Initializing the FPU 20-2
20.2.3.1 Intel486 DX CPU SOFTWARE EMULATION 20-3
20.2.3.2 Intel486 SX CPU SOFTWARE EMULATION PROCEDURE 20-3
20.2.4 Handling Numerics Exceptions 20-4
20.2.5 Simultaneous Exception Response 20-5
20.2.6 Exception Recovery Examples 20-5

CHAPTER 21
NUMERIC PROGRAMMING EXAMPLES

21.1 CONDITIONAL BRANCHING EXAMPLE 21-1
21.2 EXCEPTION HANDLING EXAMPLES 21-2
21.3 FLOATING-POINT TO ASCII CONVERSION EXAMPLES 21-7
21.3.1 Function Partitioning 21-7
21.3.2 Exception Considerations 21-7
21.3.3 Special Instructions 21-7
21.3.4 Description of Operation 21-21
21.3.5 Scaling the Value 21-22
21.3.5.1 INACCURACY IN SCALING 21-22
21.3.5.2 AVOIDING UNDERFLOW AND OVERFLOW 21-22
21.3.5.3 FINAL ADJUSTMENTS 21-23
21.3.6 Output Format 21-23
21.4 TRIGONOMETRIC CALCULATION EXAMPLES 21-23

CHAPTER 22
EXECUTING 286 AND
Intel386™ DX OR SX CPU PROGRAMS

22.1 TWO WAYS TO RUN 286 CPU TASKS 22-2
22.2 DIFFERENCES FROM 286 CPU 22-2
22.2.1 Wraparound of 286 Processor 24-Bit Physical Address Space 22-2
22.2.2 Reserved Word of Segment Descriptor 22-2

Page

22.2.3 New Segment Descriptor Type Codes 22-2
22.2.4 Restricted Semantics of LOCK Prefix 22-3
22.2.5 Additional Exceptions 22-3
22.3 DIFFERENCES FROM Intel386 CPU 22-4
22.3.1 New Flag 22-4
22.3.2 New Exception 22-4
22.3.3 New Instructions 22-4
22.3.4 New Control Register Bits 22-5
22.3.5 New Page-Table Entry Bits 22-5
22.3.6 Changes in Segment Descriptor Loads 22-5

CHAPTER 23
REAL-ADDRESS MODE

23.1 ADDRESS TRANSLATION 23-1
23.2 REGISTERS AND INSTRUCTIONS 23-2
23.3 INTERRUPT AND EXCEPTION HANDLING 23-3
23.4 ENTERING AND LEAVING REAL-ADDRESS MODE 23-4
23.4.1 Switching to Protected Mode 23-4
23.5 SWITCHING BACK TO REAL-ADDRESS MODE 23-4
23.6 REAL-ADDRESS MODE EXCEPTIONS 23-5
23.7 DIFFERENCES FROM 8086 CPU 23-5
23.8 DIFFERENCES FROM 286 CPU IN REAL-ADDRESS MODE 23-9
23.8.1 Bus Lock 23-9
23.8.2 Location of First Instruction 23-10
23.8.3 Initial Values of General Registers 23-10
23.8.4 Bus Hold 23-10
23.8.5 Math Coprocessor Differences 23-10
23.9 DIFFERENCES FROM Intel386 DX CPU IN REAL-ADDRESS MODE 23-11
23.10 PROCESSOR DETECTION CODE 23-11

CHAPTER 24
VIRTUAL-8086 MODE

24.1 EXECUTING 8086 CPU CODE 24-1
24.1.1 Registers and Instructions 24-1
24.1.2 Address Translation 24-2
24.2 STRUCTURE OF A VIRTUAL-8086 TASK 24-3
24.2.1 Paging for Virtual-8086 Tasks 24-4
24.2.2 Protection within a Virtual-8086 Task 24-5
24.3 ENTERING AND LEAVING VIRTUAL-8086 Mode 24-5
24.3.1 Transitions Through Task Switches 24-6
24.3.2 Transitions Through Trap Gates and Interrupt Gates 24-7
24.4 ADDITIONAL SENSITIVE INSTRUCTIONS 24-8
24.4.1 Emulating 8086 Operating System Calls 24-9
24.4.2 Emulating the Interrupt-Enable Flag 24-9
24.5 VIRTUAL I/O 24-9
24.5.1 I/O-Mapped I/O 24-10
24.5.2 Memory-Mapped I/O 24-10
24.5.3 Special I/O Buffers 24-10
24.6 DIFFERENCES FROM 8086 CPU 24-10
24.7 DIFFERENCES FROM 286 CPU IN REAL-ADDRESS MODE 24-13
24.7.1 Privilege Level 24-14

24.7.2 Bus Lock .....
24.8 DIFFERENCES

CHAPTER 25
MIXING 16-BIT AND

25.1 USING 16-BIT A
25.2 MIXING 16-BIT
25.3 SHARING DATA
25.4 TRANSFERRIN
25.4.1 Size of Code-S
25.4.2 Stack Manage
25.4.2.1 CONTROLL
25.4.2.2 CHANGING
25.4.3 Interrupt Contr
25.4.4 Parameter Tra
25.4.5 The Interface

CHAPTER 26
COMPATIBILITY WIT
Intel287™ AND Intel3

26.1 DIFFERENCES
26.2 DIFFERENCES
26.2.1 Data Types ar
26.2.2 Tag, Status, a
26.2.3 Instruction Se
26.3 DIFFERENCES

CHAPTER 27
INSTRUCTION SET

27.1 OPERAND-SIZ
27.1.1 Default Segm
27.1.2 Operand-Size
27.1.3 Address-Size
27.2 INSTRUCTION
27.2.1 ModR/M and
27.2.2 How to Read
27.2.2.1 OPCODE C
27.2.2.2 INSTRUCT
27.2.2.3 CLOCKS C
27.2.2.4 DESCRIPT
27.2.2.5 OPERATIO
27.2.2.6 DESCRIPT
27.2.2.7 FLAGS AF
27.2.2.8 PROTECTE
27.2.2.9 REAL ADD
27.2.2.10 VIRTUAL-
AAA .....
AAD .....
AAM .....

Page

24.7.2 Bus Lock 24-14
24.8 DIFFERENCES FROM Intel386™ DX AND SX CPUs 24-15

CHAPTER 25
MIXING 16-BIT AND 32-BIT CODE

25.1 USING 16-BIT AND 32-BIT ENVIRONMENTS 25-1
25.2 MIXING 16-BIT AND 32-BIT OPERATIONS 25-2
25.3 SHARING DATA AMONG MIXED-SIZE CODE SEGMENTS 25-3
25.4 TRANSFERRING CONTROL AMONG MIXED-SIZE CODE SEGMENTS 25-4
25.4.1 Size of Code-Segment Pointer 25-4
25.4.2 Stack Management for Control Transfers 25-4
25.4.2.1 CONTROLLING THE OPERAND SIZE FOR A CALL 25-6
25.4.2.2 CHANGING SIZE OF A CALL 25-6
25.4.3 Interrupt Control Transfers 25-6
25.4.4 Parameter Translation 25-7
25.4.5 The Interface Procedure 25-7

CHAPTER 26
COMPATIBILITY WITH THE 8087,
Intel287™ AND Intel387™ MATH COPROCESSORS

26.1 DIFFERENCES FROM Intel386 CPU/Intel387 NPX SYSTEMS 26-1
26.2 DIFFERENCES FROM 286/Intel287 SYSTEMS 26-2
26.2.1 Data Types and Exception Handling 26-3
26.2.2 Tag, Status, and Control Words 26-6
26.2.3 Instruction Set 26-8
26.3 DIFFERENCES FROM 8086/8087 SYSTEMS 26-11

CHAPTER 27
INSTRUCTION SET

27.1 OPERAND-SIZE AND ADDRESS-SIZE ATTRIBUTES 27-1
27.1.1 Default Segment Attribute 27-1
27.1.2 Operand-Size and Address-Size Instruction Prefixes 27-1
27.1.3 Address-Size Attribute for Stack 27-2
27.2 INSTRUCTION FORMAT 27-2
27.2.1 ModR/M and SIB Bytes 27-3
27.2.2 How to Read the Instruction Set Pages 27-8
27.2.2.1 OPCODE COLUMN 27-8
27.2.2.2 INSTRUCTION COLUMN 27-9
27.2.2.3 CLOCKS COLUMN 27-11
27.2.2.4 DESCRIPTION COLUMN 27-12
27.2.2.5 OPERATION 27-12
27.2.2.6 DESCRIPTION 27-16
27.2.2.7 FLAGS AFFECTED 27-16
27.2.2.8 PROTECTED MODE EXCEPTIONS 27-17
27.2.2.9 REAL ADDRESS MODE EXCEPTIONS 27-17
27.2.2.10 VIRTUAL-8086 MODE EXCEPTIONS 27-17
AAA 27-18
AAD 27-19
AAM 27-20

	Page
AAS	27-21
ADC	27-22
ADD	27-24
AND	27-25
ARPL	27-26
BOUND	27-28
BSF	27-30
BSR	27-32
BSWAP	27-34
BT	27-35
BTC	27-37
BTR	27-39
BTS	27-41
CALL	27-43
CBW/CWDE	27-50
CLC	27-51
CLD	27-52
CLI	27-53
CLTS	27-54
CMC	27-55
CMP	27-56
CMPS/CMPSB/CMPSW/CMPSD	27-57
CMPXCHG	27-60
CPUID	27-62
CWD/CDQ	27-65
DAA	27-66
DAS	27-67
DEC	27-68
DIV	27-69
ENTER	27-71
F2XM1	27-73
FABS	27-75
FADD/FADDP/FIADD	27-76
FBLD	27-78
FBSTP	27-80
FCHS	27-81
FCLEX/FNCLEX	27-82
FCOM/FCOMP/FCOMPP	27-83
FCOS	27-85
FDECSTP	27-87
FDIV/FDIVP/FIDIV	27-88
FDIVR/FDIVPR/FIDIVR	27-90
FFREE	27-92
FICOM/FICOMP	27-93
FILD	27-95
FINCSTP	27-97
FINIT/FNINIT	27-98
FIST/FISTP	27-100
FLD	27-102
FLD1/FLDL2T/FLDL2E/ FLDPI/FLDLG2/FLDLN2/FLDZ	27-104
FLDCW	27-106
FLDENV	27-107

FMUL/FMULP/FIMUL
FNOP
FPATAN
FPREM
FPREM1
FPTAN
FRNDINT
FRSTOR
FSAVE/FNSAVE
FSCALE
FSIN
FSINCOS
FSQRT
FST/FSTP
FSTCW/FNSTCW
FSTENV/FNSTENV
FSTSW/FNSTSW
FSUB/FSUBP/FISUB
FSUBR/FSUBPR/FIS
FTST
FUCOM/FUCOMP/FU
FWAIT
FXAM
FXCH
FXTRACT
FYL2X
FYL2XP1
HLT
IDIV
IMUL
IN
INC
INS/INSB/INSW/INS
INT/INTO
INVD
INVLPG
IRET/IRETD
JCC
JMP
LAHF
LAR
LEA
LEAVE
LGDT/LIDT
LGS/LSS/LDS/LES
LLDT
LMSW
LOCK
LODS/LODSB/LOD
LOOP/LOOPcond
LSL
LTR
MOV

	Page
FMUL/FMULP/FIMUL	27-109
FNOP	27-111
FPATAN	27-112
FPREM	27-114
FPREM1	27-116
FPTAN	27-118
FRNDINT	27-120
FRSTOR	27-121
FSAVE/FNSAVE	27-123
FSCALE	27-125
FSIN	27-126
FSINCOS	27-128
FSQRT	27-130
FST/FSTP	27-131
FSTCW/FNSTCW	27-133
FSTENV/FNSTENV	27-135
FSTSW/FNSTSW	27-137
FSUB/FSUBP/FISUB	27-139
FSUBR/FSUBPR/FISUBR	27-141
FTST	27-143
FUCOM/FUCOMP/FUCOMPP	27-145
FWAIT	27-147
FXAM	27-148
FXCH	27-150
FXTRACT	27-151
FYL2X	27-153
FYL2XP1	27-155
HLT	27-157
IDIV	27-158
IMUL	27-160
IN	27-162
INC	27-163
INS/INSB/INSW/INSD	27-164
INT/INTO	27-166
INVD	27-171
INVLPG	27-173
IRET/IRETD	27-174
JCC	27-179
JMP	27-183
LAHF	27-188
LAR	27-189
LEA	27-191
LEAVE	27-193
LGDT/LIDT	27-194
LGS/LSS/LDS/LES/LFS	27-196
LLDT	27-199
LMSW	27-201
LOCK	27-202
LODS/LODSB/LODSW/LODSD	27-204
LOOP/LOOPcond	27-206
LSL	27-208
LTR	27-210
MOV	27-211

	Page
MOVS/MOVSB/MOVSW/MOVSD	27-215
MOVSX	27-217
MOVZX	27-218
MUL	27-219
NEG	27-221
NOP	27-222
NOT	27-223
OR	27-224
OUT	27-226
OUTS/OUTSB/OUTSW/OUTSD	27-228
POP	27-230
POPA/POPAD	27-233
POPF/POPFD	27-235
PUSH	27-236
PUSHA/PUSHAD	27-238
PUSHF/PUSHFD	27-240
RCL/RCR/ROL/ROR	27-241
REP/REPE/REPZ/REPNE/REPNZ	27-244
RET	27-247
RSM	27-251
SAHF	27-253
SAL/SAR/SHL/SHR	27-254
SBB	27-257
SCAS/SCASB/SCASW/SCASD	27-259
SETcc	27-261
SGDT/SIDT	27-263
SHLD	27-265
SHRD	27-267
SLDT	27-269
SMSW	27-270
STC	27-271
STD	27-272
STI	27-273
STOS/STOSB/STOSW/STOSD	27-274
STR	27-276
SUB	27-277
TEST	27-279
VERR, VERW	27-280
WAIT	27-282
WBINVD	27-283
XADD	27-285
XCHG	27-286
XLAT/XLATB	27-288
XOR	27-290

APPENDIX A
OPCODE MAP

APPENDIX B
FLAG CROSS-REFERENCE

APPENDIX C
STATUS FLAG SUMMARY

APPENDIX D
CONDITION CODES

APPENDIX E
INSTRUCTION FORM

APPENDIX F
NUMERIC EXCEPTIO

APPENDIX G
CODE OPTIMIZATIO

APPENDIX H
ADVANCED FEATUR

APPENDIX I
REVISION HISTORY

Figure

1-1	Bit and By
2-1	Segmente
2-2	Fundame
2-3	Bytes, W
2-4	Data Typ
2-5	Applicatio
2-6	An Unse
2-7	A Segme
2-8	Stacks
2-9	EFLAGS
2-10	Effective
3-1	PUSH In
3-2	PUSHA
3-3	POP Ins
3-4	POPA In
3-5	Sign Ex
3-6	SHL/SA
3-7	SHR Ins
3-8	SAR Ins
3-9	SHLD I
3-10	SHRD I
3-11	ROL Ins
3-12	ROR In
3-13	RCL Ins
3-14	RCR In
3-15	Formal
3-16	Nested
3-17	Stack I
3-18	Stack I
3-19	Stack I

APPENDIX D
CONDITION CODES

APPENDIX E
INSTRUCTION FORMAT AND TIMING

APPENDIX F
NUMERIC EXCEPTION SUMMARY

APPENDIX G
CODE OPTIMIZATION

APPENDIX H
ADVANCED FEATURES

APPENDIX I
REVISION HISTORY

Figures

Figure	Title	Page
1-1	Bit and Byte Order	1-7
2-1	Segmented Addressing	2-4
2-2	Fundamental Data Types	2-5
2-3	Bytes, Words, and Doublewords in Memory	2-5
2-4	Data Types	2-7
2-5	Application Register Set	2-9
2-6	An Unsegmented Memory	2-11
2-7	A Segmented Memory	2-12
2-8	Stacks	2-13
2-9	EFLAGS Register	2-20
2-10	Effective Address Computation	3-2
3-1	PUSH Instruction	3-2
3-2	PUSHA Instruction	3-3
3-3	POP Instruction	3-4
3-4	POPA Instruction	3-5
3-5	Sign Extension	3-5
3-6	SHL/SAL Instruction	3-14
3-7	SHR Instruction	3-14
3-8	SAR Instruction	3-15
3-9	SHLD Instruction	3-16
3-10	SHRD Instruction	3-17
3-11	ROL Instruction	3-18
3-12	ROR Instruction	3-18
3-13	RCL Instruction	3-18
3-14	RCR Instruction	3-30
3-15	Formal Definition of the ENTER Instruction	3-31
3-16	Nested Procedures	3-32
3-17	Stack Frame After Entering MAIN	3-33
3-18	Stack Frame After Entering PROCEDURE A	3-34
3-19	Stack Frame After Entering PROCEDURE B	3-35

Figures

Figure	Title	Page
3-20	Stack Frame After Entering PROCEDURE C	1-7
3-21	Low Byte of EFLAGS Register	3-37
3-22	Flags Used with PUSHF and POPF	3-42
3-23	ASCII Arithmetic Using BSWAP	3-43
3-23	ASCII Arithmetic Using BSWAP (Contd.)	4-2
4-1	System Flags	4-2
4-2	Memory Management Registers	4-4
4-3	Control Registers	4-6
4-4	Debug Registers	4-9
4-5	Test Registers	4-10
4-6	Cache Test Registers (All Intel486™ Processors Except the IntelDX4™ Processor and Write-Back Enhanced IntelDX4 Processor)	4-11
4-7	IntelDX4™ Processor and Write-Back Enhanced IntelDX4 Processor Cache Test Registers	4-11
4-8	TR4 and TR5 Definition in Write-Back Mode for the Write-Back Enhanced IntelDX2 Processor	4-12
4-9	TR4 and TR5 Definition in Write-Back Mode for the Write-Back Enhanced IntelDX4 Processor	5-3
5-1	Flat Model	5-3
5-2	Protected Flat Model	5-4
5-3	Multi-Segment Model	5-5
5-4	TI Bit Selects Descriptor Table	5-7
5-5	Segment Translation	5-8
5-6	Segment Registers	5-8
5-7	Segment Selector	5-9
5-8	Segment Descriptors	5-14
5-9	Segment Descriptor (Segment Not Present)	5-15
5-10	Descriptor Tables	5-16
5-11	Pseudo-Descriptor Format	5-18
5-12	Format of a Linear Address	5-19
5-13	Page Translation	5-20
5-14	Format of a Page Table Entry	5-20
5-15	Format of a Page Table Entry for a Not-Present Page	5-23
5-16	Combined Segment and Page Address Translation	5-24
5-17	Each Segment Can Have Its Own Page Table	6-2
6-1	Descriptor Fields Used for Protection (Part 1 of 2)	6-2
6-1	Descriptor Fields Used for Protection (Part 2 of 2)	6-3
6-2	Protection Rings	6-6
6-3	Privilege Check for Data Access	6-8
6-4	Privilege Check for Control Transfer Without Gate	6-9
6-5	Call Gate	6-12
6-6	Call Gate Mechanism	6-13
6-7	Privilege Check for Control Transfer with Call Gate	6-15
6-8	Initial Stack Pointers in a TSS	6-16
6-9	Stack Frame During Interlevel Call	6-22
6-10	Protection Fields of a Page Table Entry	7-3
7-1	Task State Segment	7-3
7-2	TSS Descriptor	7-4

Figure	Title
7-3	TR Register
7-4	Task Gate D
7-5	Task Gates
7-6	Nested Task
7-7	Overlapping
8-1	Memory-Ma
8-2	I/O Permiss
9-1	IDTR Regis
9-2	IDT Gate D
9-3	Interrupt Pr
9-4	Stack Fram
9-5	Interrupt Ta
9-6	Error Code
9-7	Page Fault
10-1	Contents o
10-2	Contents o
10-3	TLB Structu
10-4	TLB Test R
10-5	Cache Stru
10-6	Cache Tes processor
10-7	IntelDX4™ Cache Tes
10-8	TR4 and T Enhanced
10-9	TR4 and T Enhanced
11-1	Debug Re
14-1	Redirectin
14-2	Transition
14-3	SMM Rev
14-4	Halt Auto
14-5	SMM Bas
14-6	SMRAM
15-1	Evolution
16-1	Intel486™
16-2	Intel486™
16-3	Intel486™
16-4	Tag Word
16-5	Opcode
16-6	Protected Memory,
16-7	Real Mod 32-Bit Fo
16-8	Protected Memory,
16-9	Real Mo 16-Bit Fo
16-10	Double-F

Figures

Figure	Title	Page
7-3	TR Register	1-7
7-4	Task Gate Descriptor	7-6
7-5	Task Gates Reference Tasks	7-8
7-6	Nested Tasks	7-14
7-7	Overlapping Linear-to-Physical Mappings	8-3
8-1	Memory-Mapped I/O	8-3
8-2	I/O Permission Bit Map	8-7
9-1	IDTR Register Locates IDT in Memory	9-6
9-2	IDT Gate Descriptors	9-8
9-3	Interrupt Procedure Call	9-9
9-4	Stack Frame After Exception or Interrupt	9-12
9-5	Interrupt Task Switch	9-13
9-6	Error Code	9-22
9-7	Page Fault Error Code	10-2
10-1	Contents of the EDX Register After Reset	10-2
10-2	Contents of the CR0 Register After Reset	10-2
10-3	TLB Structure	10-9
10-4	TLB Test Registers	10-13
10-5	Cache Structure	10-14
10-6	Cache Test Registers (All Intel486™ processors except the IntelDX4™ processor and the Write-Back Enhanced IntelDX4 processor)	10-15
10-7	IntelDX4™ Processor and Write-Back Enhanced IntelDX4 Processor Cache Test Registers	10-17
10-8	TR4 and TR5 Definition for Write-Back Mode for the Write-Back Enhanced IntelDX2 Processor	10-17
10-9	TR4 and TR5 Definition for Write-Back Mode for the Write-Back Enhanced IntelDX4 Processor	11-3
11-1	Debug Registers	11-3
14-1	Redirecting System Memory Addresses to SMRAM	14-3
14-2	Transition to and from System Management Mode	14-6
14-3	SMM Revision Identifier	14-10
14-4	Halt Auto Retart	14-11
14-5	SMM Base Relocation	14-12
14-6	SMRAM Usage	15-2
15-1	Evolution and Performance of Numeric Processors	15-2
16-1	Intel486™ FPU Register Set	16-2
16-2	Intel486™ FPU Status Word	16-3
16-3	Intel486™ FPU Control Word Format	16-6
16-4	Tag Word Format	16-7
16-5	Opcode Field	16-7
16-6	Protected Mode Numeric Instruction and Data Pointer Image in Memory, 32-Bit Format	16-8
16-7	Real Mode Numeric Instruction and Data Pointer Image in Memory, 32-Bit Format	16-9
16-8	Protected Mode Numeric Instruction and Data Pointer Image in Memory, 16-Bit Format	16-9
16-9	Real Mode Numeric Instruction and Data Pointer Image in Memory, 16-Bit Format	16-11
16-10	Double-Precision Number System	16-12

Figures

Figure	Title	Page
16-11	Numerical Data Formats	1-7
17-1	Floating-Point System with Denormals	17-5
17-2	Floating-Point System without Denormals	17-5
17-3	Arithmetic Example Using Infinity	17-23
17-4	Coprocessor Detection Code	19-2
19-1	Sample C-386/486 Program	19-2
19-2	Sample Numeric Constants	19-5
19-3	Status Word Record Definition	19-6
19-4	Structure Definition	19-6
19-5	Sample PL/M-386/486 Program	19-8
19-6	Sample ASM386/486 Program	19-9
19-6	Sample ASM386/486 Program (Contd.)	19-11
19-7	Instructions and Register Stack	19-13
19-8	Exception Synchronization Examples	19-15
19-9	The two cases that arise depending on the timing of the receipt of the external interrupt.	21-2
21-1	Conditional Branching for Compares	21-2
21-2	Conditional Branching for FXAM	21-3
21-3	Full-State Exception Handler	21-4
21-4	Reduced-Latency Exception Handler	21-5
21-5	Reentrant Exception Handler	21-6
21-6	Floating-Point to ASCII Conversion Routine	21-8
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-9
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-11
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-12
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-13
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-14
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-15
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-16
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-17
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-18
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-19
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-20
21-6	Floating-Point to ASCII Conversion Routine (Contd.)	21-24
21-7	Relationships Between Adjacent Joints	21-26
21-8	Robot Arm Kinematics Example	21-27
21-8	Robot Arm Kinematics Example (Contd.)	21-28
21-8	Robot Arm Kinematics Example (Contd.)	21-29
21-8	Robot Arm Kinematics Example (Contd.)	21-30
21-8	Robot Arm Kinematics Example (Contd.)	21-31
21-8	Robot Arm Kinematics Example (Contd.)	21-32
21-8	Robot Arm Kinematics Example (Contd.)	21-33
21-8	Robot Arm Kinematics Example (Contd.)	23-2
23-1	8086 Address Translation	23-2
23-2	Real-Address Detection Code	24-3
24-1	8086 Address Translation	24-3
24-2	Entering and Leaving Virtual-8086 Mode	24-6
24-3	Privilege Level 0 Stack After Interrupt in Virtual-8086 Mode	24-7
25-1	Stack After Far 16- and 32-Bit Calls	25-5

Figure

27-1	Intel486™ F
27-2	ModR/M an
27-3	Bit Offset fo
27-4	Memory Bit

Table

2-1	Register Na
2-2	Status Flag
2-3	Default Seg
2-4	Exceptions
3-1	Operands f
3-2	Bit Test and
3-3	Conditional
3-4	Repeat Ins
3-5	Flag Contro
5-1	Application
6-1	System Se
6-2	Interlevel F
6-3	Valid Desc
6-4	Combined
7-1	Checks Ma
7-2	Effect of a
9-1	Exception
9-2	Priority Am
9-3	Intel Rese
9-4	Interrupt a
9-6	Alignment
9-7	Exception
9-8	Error Code
10-1	Processor
10-2	Recomme processor
10-3	EM and M
10-4	Meaning o
10-6	State Bit A Write-Bac
10-5	Encoding
11-1	Breakpoin
11-2	Debug Ex
12-2	Write-Bac processor
12-1	Cache Op and Write
14-1	SMRAM S
14-3	SMM Initi

Tables

Table	Title	Page
15-1	Numeric Processing Speed Comparisons	2-8
15-2	Numeric Data Types	15-6
15-3	Principal Numeric Instructions	15-7
16-1	Condition Code Interpretation	16-4
16-2	Correspondence Between FPU and IU Flag Bits	16-5
16-3	Summary of Format Parameters	16-14
16-4	Real Number Notation	16-16
16-5	Rounding Modes	17-2
17-1	Arithmetic and Nonarithmetic Instructions	17-2
17-2	Denormalized Values	17-3
17-3	Zero Operands and Results	17-6
17-3	Zero Operands and Results	17-7
17-4	Infinity Operands and Results	17-8
17-4	Infinity Operands and Results	17-9
17-4	Infinity Operands and Results	17-12
17-5	Rules for Generating QNaNs	17-13
17-6	Binary Integer Encodings	17-14
17-7	Packed Decimal Encodings	17-15
17-8	Single and Double Real Encodings	17-16
17-9	Extended Real Encodings	17-17
17-10	Unsupported Formats	17-21
17-11	Masked Responses to Invalid Operations	17-24
17-12	Masked Overflow Results	18-2
18-1	Data Transfer Instructions	18-2
18-2	Nontranscendental Instructions (Besides Basic Arithmetic)	18-3
18-3	Basic Arithmetic Instructions and Operands	18-3
18-4	Comparison Instructions	18-4
18-5	TEST Constants for Conditional Branching	18-5
18-6	Transcendental Instructions	18-5
18-8	Control Instructions	18-7
19-1	PL/M-386/486 Built-In Procedures	19-3
19-2	ASM386/486 Storage Allocation Directives	19-4
19-3	Addressing Method Examples	19-7
20-1	FPU State Following Initialization	20-3
23-1	Exceptions and Interrupts	23-6
23-2	New Intel486™ CPU Exceptions	23-9
27-1	Effective Size Attributes	27-2
27-2	16-Bit Addressing Forms with the ModR/M Byte	27-5
27-3	32-Bit Addressing Forms with the ModR/M Byte	27-6
27-4	32-Bit Addressing Forms with the SIB Byte	27-7
27-5	Task Switch Times for Exceptions	27-17
27-6	Exceptions	27-17
A-1	Opcodes determined by bits 5,4,3 of modR/M byte:	A-8
A-2	Opcodes determined by bits 5,4,3 of modR/M byte:	A-8

Introduc
Process